BCS547

Unsupervised Learning

Hebb Rule

- Linear neuron

$$v = \mathbf{w}^T \mathbf{u}$$

- Hebb rule

$$\tau_w \frac{d\mathbf{w}}{dt} = v\mathbf{u}, \quad \mathbf{w}_{t+1} = \mathbf{w}_t + \frac{\delta t}{\tau_w} v\mathbf{u}$$

- Similar to LTP (but not quite…)

Hebb Rule

- Average Hebb rule= correlation rule

$$\tau_w \frac{d\mathbf{w}}{dt} = v\mathbf{u}$$

$$\tau_w \frac{d\mathbf{w}}{dt} = \langle v\mathbf{u} \rangle = \left\langle \left(\mathbf{w}^T\mathbf{u}\right)\mathbf{u} \right\rangle = \left\langle \mathbf{u}\left(\mathbf{w}^T\mathbf{u}\right) \right\rangle$$

$$= \left\langle \mathbf{u}\left(\mathbf{u}^T\mathbf{w}\right) \right\rangle = \left\langle \mathbf{u}\mathbf{u}^T \right\rangle \mathbf{w} = \mathbf{Q}\mathbf{w}$$

- Q: *correlation* matrix of **u**

Hebb Rule

- Hebb rule with threshold= covariance rule

$$\tau_w \frac{d\mathbf{w}}{dt} = \left\langle v\left(\mathbf{u} - \langle \mathbf{u} \rangle\right)\right\rangle = \left\langle \left(\mathbf{w}^T \mathbf{u}\right)\left(\mathbf{u} - \langle \mathbf{u} \rangle\right)\right\rangle$$

$$= C\mathbf{w}$$

$$C = \left\langle \left(\mathbf{u} - \langle \mathbf{u} \rangle\right)\left(\mathbf{u} - \langle \mathbf{u} \rangle\right)^T \right\rangle = \left\langle \left(\mathbf{u} - \langle \mathbf{u} \rangle\right)\mathbf{u}^T \right\rangle$$

- C: *covariance* matrix of **u**
- Note that <(**v**-< **v** >)(**u**-< **u** >)> would be unrealistic because it predicts LTP when both **u** and **v** are low

Hebb Rule

- Main problem with Hebb rule: it's unstable… Two solutions:
  1. Bounded weights
  2. Normalization of either the activity of the postsynaptic cells or the weights.

BCM rule

- Hebb rule with sliding threshold

$$\tau_w \frac{d\mathbf{w}}{dt} = v\left(v - \theta_v\right)\mathbf{u}$$

$$\tau_{\theta_v} \frac{d\theta_v}{dt} = v^2 - \theta_v$$

$$\tau_w < \tau_{\theta_v}$$

- BCM rule implements competition because when a synaptic weight grows, it raises by v^2, making more difficult for other weights to grow.

Weight Normalization

- Subtractive Normalization: $\sum_{i=1}^{N_u} w_i = \mathbf{n} \cdot \mathbf{w} = \text{Const.}$

$$\tau_w \frac{d\mathbf{w}}{dt} = v\mathbf{u} - \frac{v(\mathbf{n} \cdot \mathbf{u})\mathbf{n}}{N_u}, \quad \mathbf{n} = [11 \cdots 11]$$

$$\tau_w \frac{dw_i}{dt} = vu_i - \frac{1}{N_u} \sum_{k=1}^{N_u} vu_k$$

$$\tau_w \frac{d\mathbf{n} \cdot \mathbf{w}}{dt} = \mathbf{n} \cdot \left(v\mathbf{u} - \frac{v(\mathbf{n} \cdot \mathbf{u})\mathbf{n}}{N_u} \right)$$

$$= v\mathbf{n} \cdot \mathbf{u} \left(1 - \frac{\mathbf{n} \cdot \mathbf{n}}{N_u} \right) = 0$$

Weight Normalization

- Multiplicative Normalization: $\sum_{i=1}^{N_u} w_i^2 = \text{Const.}$

$$\tau_w \frac{d\mathbf{w}}{dt} = v\mathbf{u} - \alpha v^2 \mathbf{w}$$

$$\tau_w \frac{d\left\|\mathbf{w}\right\|^2}{dt} = \tau_w 2\mathbf{w} \cdot \frac{d\mathbf{w}}{dt}$$

$$\frac{d\left\|\mathbf{w}\right\|^2}{dt} = 2v^2\left(1 - \alpha\left\|\mathbf{w}\right\|^2\right)$$

- Norm of the weights converge to $1/\alpha$

Hebb Rule

- Convergence properties:

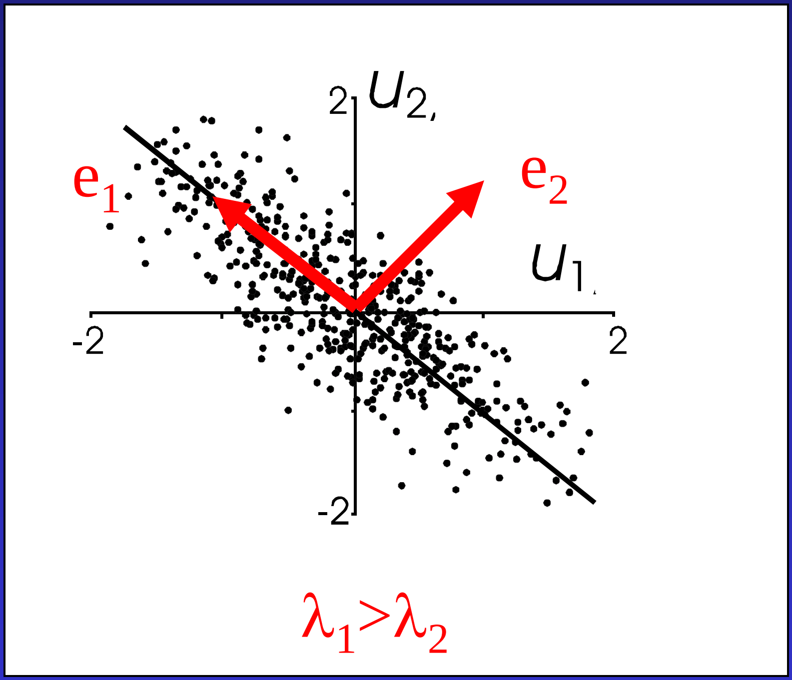
$$\tau_w \frac{d\mathbf{w}}{dt} = \mathbf{Q}\mathbf{w}$$

- Use an eigenvector decomposition:

$$\mathbf{w}(t) = \sum_{\mu=1}^{N_u} c_\mu(t)\,\mathbf{e}_\mu$$

where $\mathbf{e}_\mu$ are the eigenvectors of $\mathbf{Q}$

Hebb Rule



Hebb Rule

$$\tau_w \frac{d\mathbf{w}(t)}{dt} = \mathbf{Q}\mathbf{w}(t), \ \mathbf{w}(t) = \sum_{\mu=1}^{N_u} c_\mu(t)\mathbf{e}_\mu$$

$$\tau_w \frac{dc_\mu(t)\mathbf{e}_\mu}{dt} = \mathbf{Q}c_\mu(t)\mathbf{e}_\mu$$

→ Equations decouple because $\mathbf{e}_\mu$ *are* the eigenvectors of Q

$$\tau_w \frac{dc_\mu(t)}{dt}\mathbf{e}_\mu = c_\mu(t)\mathbf{Q}\mathbf{e}_\mu$$

$$\tau_w \frac{dc_\mu(t)}{dt}\mathbf{e}_\mu = c_\mu(t)\lambda_\mu\mathbf{e}_\mu$$

Hebb Rule

$$\tau_w \frac{dc_\mu(t)}{dt}\mathbf{e}_\mu = c_\mu(t)\lambda_\mu \mathbf{e}_\mu$$

$$\tau_w \frac{dc_\mu(t)}{dt} = c_\mu(t)\lambda_\mu$$

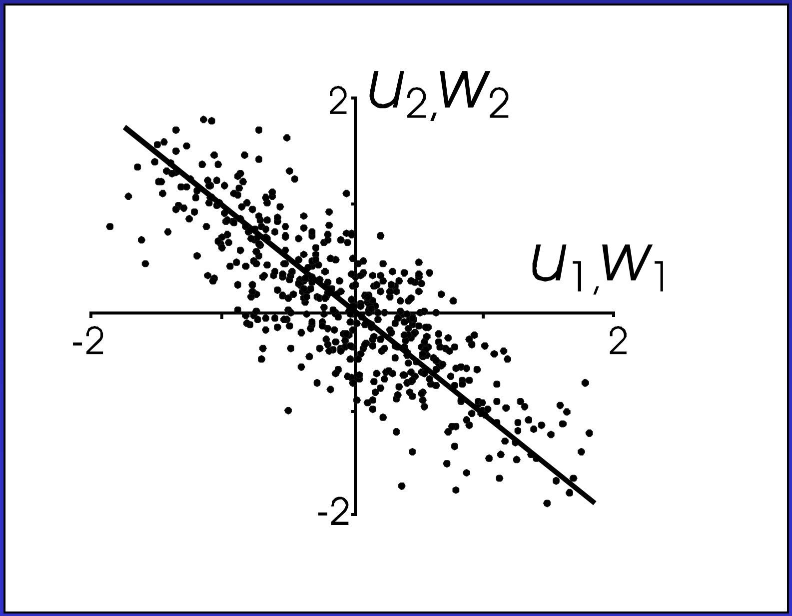
$$c_\mu(t) = \exp\left(\frac{\lambda_\mu t}{\tau_w}\right)c_\mu(0) = \exp\left(\frac{\lambda_\mu t}{\tau_w}\right)\left(w(0)\cdot \mathbf{e}_\mu\right)$$

$$\mathbf{w}(t) = \sum_{\mu=1}^{N_u} \exp\left(\frac{\lambda_\mu t}{\tau_w}\right)\left(w(0)\cdot \mathbf{e}_\mu\right)\mathbf{e}_\mu$$

for large t, $\mathbf{w}(t) \propto \mathbf{e}_1,\ v \propto \mathbf{e}_1 \cdot \mathbf{u}$

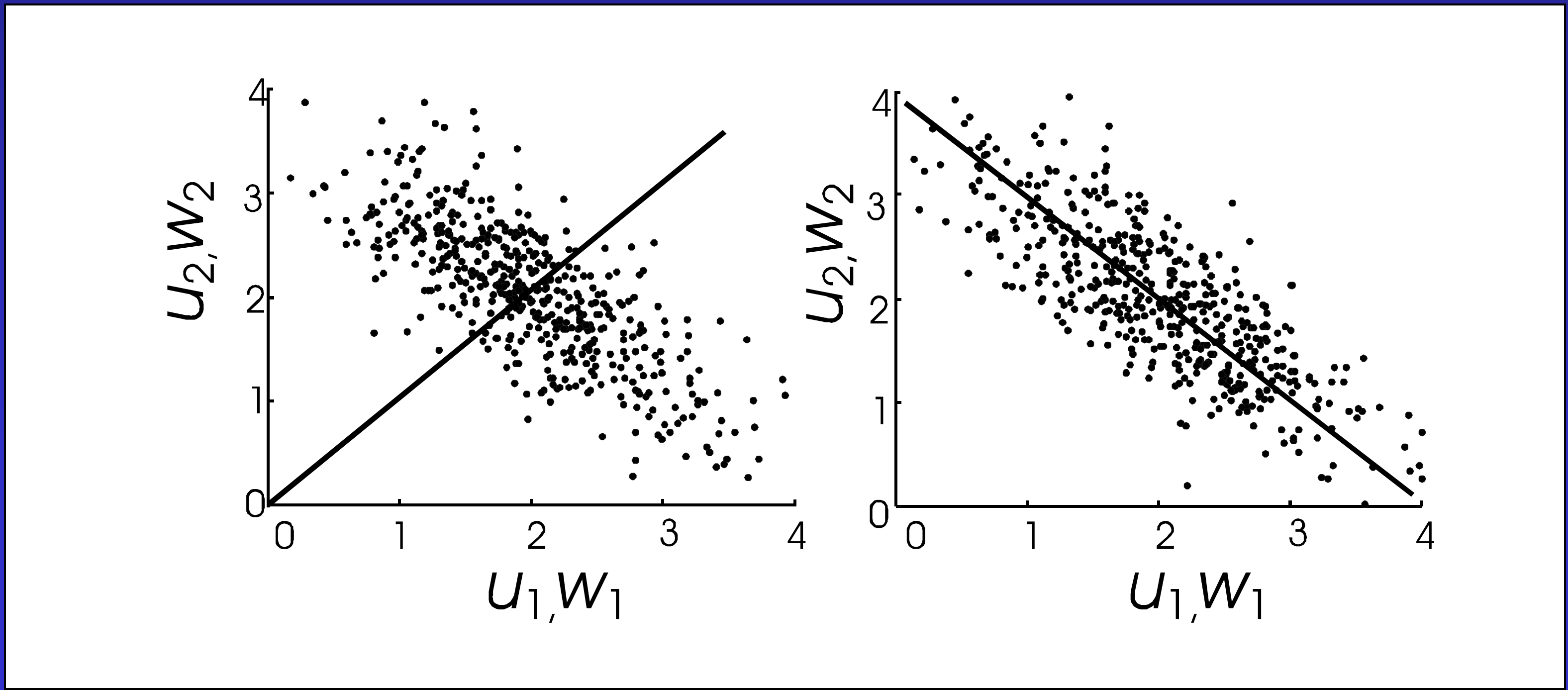
Hebb Rule

- The weights line up with first eigenvector and the postsynaptic activity, *v*, converges toward the projection of **u** onto the first eigenvector (unstable PCA)



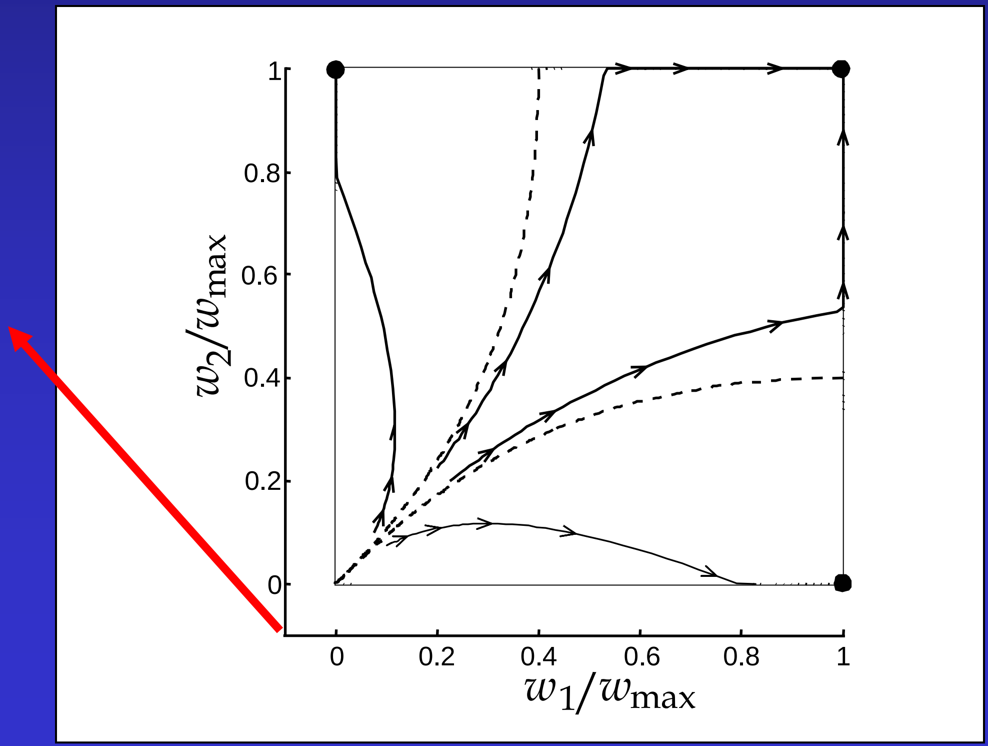
Hebb Rule

- Non zero mean distribution: correlation vs covariance



Hebb Rule

- Limiting weights growth affects the final state



First eigenvector: [1,-1]

Hebb Rule

- Normalization also affects the final state.
- Ex: multiplicative normalization. In this case, Hebb rule extracts the first eigenvector but keeps the norm constant (stable PCA).

Hebb Rule

- Normalization also affects the final state.
- Ex: subtractive normalization.

$$\tau_w \frac{d\mathbf{w}}{dt} = \mathbf{Q}\mathbf{w} - \frac{(\mathbf{w}\cdot\mathbf{Q}\mathbf{n})\mathbf{n}}{N_u}$$

$$\text{if} \;\; \mathbf{e}_1 \propto \mathbf{n}$$

$$\tau_w \frac{d\mathbf{e}_1}{dt} = \mathbf{Q}\mathbf{e}_1 - \frac{(\mathbf{e}_1\cdot\mathbf{Q}\mathbf{n})\mathbf{n}}{N_u} = \mathbf{Q}\mathbf{n} - \frac{(\mathbf{n}\cdot\mathbf{Q}\mathbf{n})\mathbf{n}}{N_u} = 0$$

Hebb Rule

$$\text{if } \mathbf{e}_1 \propto \mathbf{n}$$

$$\mathbf{e}_\mu \cdot \mathbf{n} = 0, \forall \mu \neq 1$$

$$\tau_w \frac{dc_\mu(t)}{dt}\mathbf{e}_\mu = \mathbf{Q}c_\mu(t)\mathbf{e}_\mu - \frac{\left(c_\mu(t)\mathbf{e}_\mu \cdot \mathbf{Qn}\right)\mathbf{n}}{N_u}$$

$$= \mathbf{Q}c_\mu(t)\mathbf{e}_\mu - \frac{\left(c_\mu(t)\mathbf{e}_\mu \cdot \mathbf{Qe}_1\right)\mathbf{n}}{N_u}$$

$$= \mathbf{Q}c_\mu(t)\mathbf{e}_\mu - \frac{\left(c_\mu(t)\mathbf{e}_\mu \cdot \lambda_1\mathbf{e}_1\right)\mathbf{n}}{N_u}$$

$$= \mathbf{Q}c_\mu(t)\mathbf{e}_\mu$$

Hebb Rule

- The constrain does not affect the other eigenvector:

$$\mathbf{w}(t) = \left(w(0)\cdot\mathbf{e}_1\right)\mathbf{e}_1 + \sum_{\mu=2}^{N_u} \exp\left(\frac{\lambda_\mu t}{\tau_w}\right)\left(w(0)\cdot\mathbf{e}_\mu\right)\mathbf{e}_\mu$$

- The weights converge to the second eigenvector (the weights need to be bounded to guarantee stability…)

Ocular Dominance Column

- One unit with one input from right and left eyes

$$v = w_R u_R + w_L u_L$$

$$\mathbf{Q} = \langle \mathbf{u}\mathbf{u} \rangle = \begin{pmatrix} \langle u_R u_R \rangle & \langle u_L u_R \rangle \\ \langle u_R u_L \rangle & \langle u_L u_L \rangle \end{pmatrix} = \begin{pmatrix} q_s & q_d \\ q_d & q_s \end{pmatrix}$$

s: same eye

d: different eyes

Ocular Dominance Column

$$\mathbf{Q} = \left\langle \mathbf{u}\mathbf{u}^T \right\rangle = \begin{pmatrix} \langle u_R u_R \rangle & \langle u_L u_R \rangle \\ \langle u_R u_L \rangle & \langle u_L u_L \rangle \end{pmatrix} = \begin{pmatrix} q_s & q_d \\ q_d & q_s \end{pmatrix}$$

- The eigenvectors are:

$$\mathbf{e}_1 = (1,1)/\sqrt{2},\ \lambda_1 = q_s + q_d$$

$$\mathbf{e}_2 = (1,-1)/\sqrt{2},\ \lambda_2 = q_s - q_d$$

Ocular Dominance Column

- Since q_d is likely to be positive, $q_s+q_d>q_s-q_d$. As a result, the weights will converge toward the first eigenvector which mixes the right and left eye equally. No ocular dominance...

$$\mathbf{e}_1 = (1,1)/\sqrt{2},\ \lambda_1 = q_s + q_d$$

$$\mathbf{e}_2 = (1,-1)/\sqrt{2},\ \lambda_2 = q_s - q_d$$

Ocular Dominance Column

- To get ocular dominance we need subtractive normalization.

$$\mathbf{e}_1 = (1,1)/\sqrt{2},\ \lambda_1 = q_s + q_d$$
$$\mathbf{e}_2 = (1,-1)/\sqrt{2},\ \lambda_2 = q_s - q_d$$

Ocular Dominance Column

- Note that the weights will be proportional to e2 or –e2 (i.e. the right and left eye are equally likely to dominate at the end). Which one wins depends on the initial conditions.

$$\mathbf{e}_1 = (1,1)/\sqrt{2},\ \lambda_1 = q_s + q_d$$

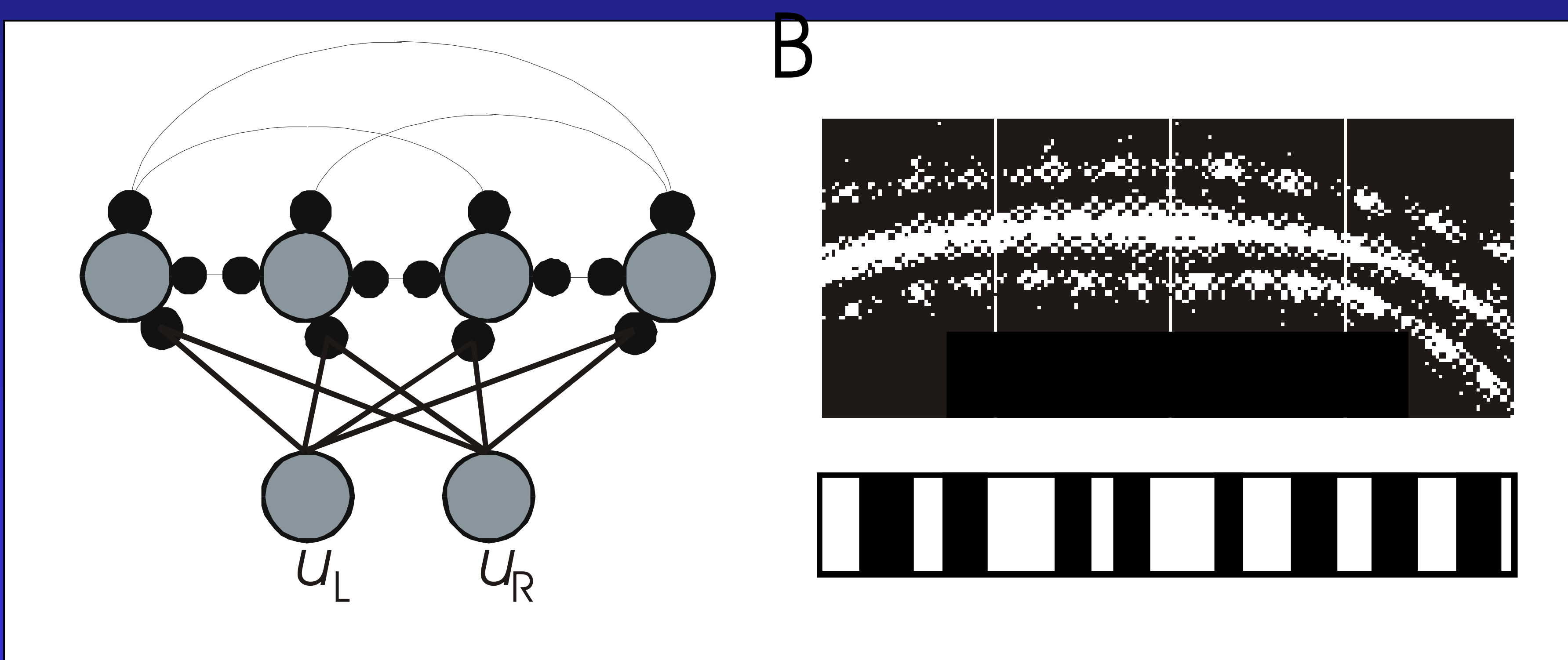
$$\mathbf{e}_2 = (1,-1)/\sqrt{2},\ \lambda_2 = q_s - q_d$$

Ocular Dominance Column

- Ocular dominance column: network with multiple output units and lateral connections.

Ocular Dominance Column

- Simplified model



Ocular Dominance Column

- If we use subtractive normalization and no lateral connections, we're back to the one cell case. Ocular dominance is determined by initial weights, i.e., it is purely stochastic. This is not what's observed in V1.
- Lateral weights could help by making sure that neighboring cells have similar ocular dominance.

Ocular Dominance Column

- Lateral weights are equivalent to feedforward weights

$$\tau \frac{dv_i}{dt} = -v_i + w_{iR}u_R + w_{iL}u_L + \mathbf{Mv}$$

$$\tau \frac{d\mathbf{v}}{dt} = -\mathbf{v} + \mathbf{w}_R u_R + \mathbf{w}_L u_L + \mathbf{Mv}$$

$$\frac{d\mathbf{v}}{dt} = 0?$$

Ocular Dominance Column

- Lateral weights are equivalent to feedforward weights

$$v_i = w_{iR} u_R + w_{iL} u_L + \mathbf{M}\mathbf{v}$$

$$\mathbf{v} = \mathbf{w}_R u_R + \mathbf{w}_L u_L + \mathbf{M}\mathbf{v}$$

$$\mathbf{v} = \left(\mathbf{I} - \mathbf{M}\right)^{-1} \left[\mathbf{w}_R \quad \mathbf{w}_L\right] \begin{bmatrix} u_R \\ u_L \end{bmatrix}$$

$$\mathbf{v} = \mathbf{K}\mathbf{W}\mathbf{u}$$

Ocular Dominance Column

$$\tau_w \frac{d\mathbf{W}}{dt} = \mathbf{v}\mathbf{u}$$

$$\mathbf{v} = \mathbf{K}\mathbf{W}\mathbf{u}$$

$$\tau_w \frac{d\mathbf{W}}{dt} = \mathbf{K}\mathbf{W}\mathbf{u}\mathbf{u}$$

$$\tau_w \frac{d\mathbf{W}}{dt} = \mathbf{K}\mathbf{W}\mathbf{Q}$$

Ocular Dominance Column

- We first project the weight vectors of each cortical unit (w_{iR}, w_{iL}) onto the eigenvectors of Q.

$$\tau_w \frac{d\mathbf{W}}{dt} = \mathbf{KWQ}$$

$$\tau_w \frac{d\mathbf{W}}{dt} = \mathbf{KWP\Lambda P}^{-1}$$

$$\tau_w \frac{d\mathbf{WP}}{dt} = \mathbf{KWP\Lambda}$$

Ocular Dominance Column

- There are two eigenvectors, $\mathbf{w}_+$ and $\mathbf{w}_-$, with eigenvalues q_s+q_d and q_s-q_d:

$$\mathbf{w}_+ = \mathbf{w}_R + \mathbf{w}_L$$

$$\mathbf{w}_- = \mathbf{w}_R - \mathbf{w}_L$$

$$\mathbf{WP} = \begin{bmatrix} \mathbf{w}_R + \mathbf{w}_L & \mathbf{w}_R - \mathbf{w}_L \end{bmatrix}$$

$$= \begin{bmatrix} \mathbf{w}_+ & \mathbf{w}_- \end{bmatrix}$$

Ocular Dominance Column

$$\tau_w \frac{d\mathbf{WP}}{dt} = \mathbf{KWP\Lambda}$$

$$\tau_w \begin{bmatrix} \dfrac{d\mathbf{w}_+}{dt} \\ \dfrac{d\mathbf{w}_-}{dt} \end{bmatrix} = \mathbf{K}\begin{bmatrix} \mathbf{w}_+ & \mathbf{w}_- \end{bmatrix} \begin{bmatrix} q_s + q_d & 0 \\ 0 & q_s - q_d \end{bmatrix}$$

$$\tau_w \begin{bmatrix} \dfrac{d\mathbf{w}_+}{dt} \\ \dfrac{d\mathbf{w}_-}{dt} \end{bmatrix} = \begin{bmatrix} \mathbf{K}(q_s + q_d)\mathbf{w}_+ \\ \mathbf{K}(q_s - q_d)\mathbf{w}_- \end{bmatrix}$$

Ocular Dominance Column

- Ocular dominance column: network with multiple output units and lateral connections.

$$\tau_w \frac{d\mathbf{w}_+}{dt} = (q_s + q_d)\mathbf{K}\mathbf{w}_+$$

$$\tau_w \frac{d\mathbf{w}_-}{dt} = (q_s - q_d)\mathbf{K}\mathbf{w}_-$$

Ocular Dominance Column

- Once again we use a subtractive normalization, which holds $\mathbf{w}_+$ constant. Consequently, the equation for $\mathbf{w}_-$ is the only one we need to worry about.

$$\tau_w \frac{d\mathbf{w}_-}{dt} = \left(q_s - q_d\right)\mathbf{K}\mathbf{w}_-$$

Ocular Dominance Column

- If the lateral weights are translation invariant, $\mathbf{Kw}_{-}$ is a convolution. This is easier to solve in the Fourier domain.

$$\tau_w \frac{d\mathbf{w}_-}{dt} = \left(q_s - q_d\right)\mathbf{Kw}_-$$

$$= \left(q_s - q_d\right)\mathbf{K}(x) * \mathbf{w}_-$$

$$\tau_w \frac{d\left(\tilde{w}_-\right)_k}{dt} = \left(q_s - q_d\right)\tilde{K}_k\left(\tilde{w}_-\right)_k$$

Ocular Dominance Column

$$\tau_w \frac{d\left(\tilde{w}_-\right)_k}{dt} = \left(q_s - q_d\right)\tilde{K}_k \left(\tilde{w}_-\right)_k$$

$$\left(\tilde{w}_-\right)_k (t) = \exp\left(\frac{\left(q_s - q_d\right)\tilde{K}_k t}{\tau_w}\right)\left(\tilde{w}_-\right)_k (0)$$

- The sine function with the highest Fourier coefficient (i.e. the fundamental) growth the fastest.

Ocular Dominance Column

- In other words, the eigenvectors of K are sine functions and the eigenvalues are the Fourier coefficients for K.

$$e_a^\mu = \cos\left(\frac{2\pi\mu a}{N_v} - \phi\right)$$

Ocular Dominance Column

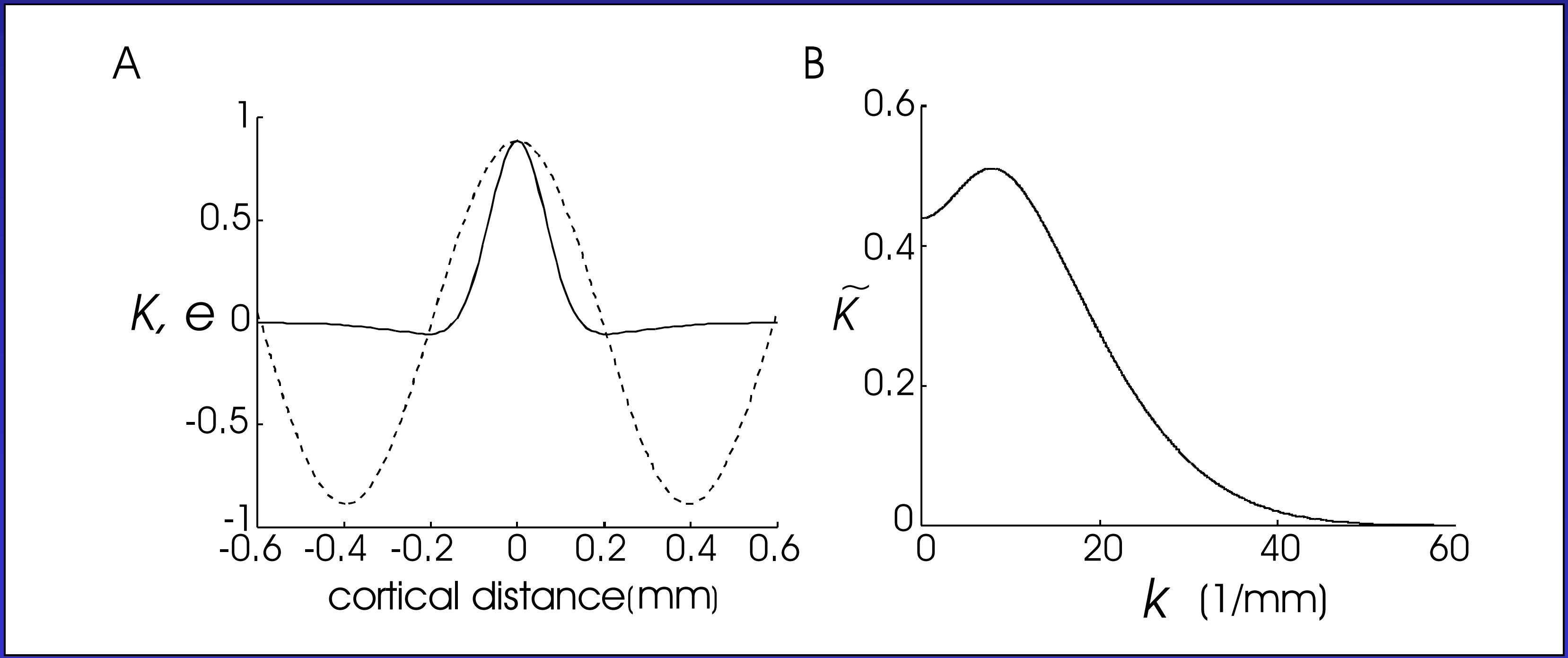
- The dynamics is dominated by the sine function with the highest Fourier coefficients, i.e., the fundamental of K(x) (note that $\mathbf{w}_-$ is not normalized along the x dimension).
- This results is an alternation of right and left columns with a periodicity corresponding to the frequency of the fundamental of K(x).

Ocular Dominance Column

- If K is a Gaussian kernel, the fundamental is the DC term and **w** ends up being constant, i.e., no ocular dominance columns (one of the eyes dominate all the cells).
- If K is a mexican hat kernel, **w** will show ocular dominance column with the same frequency as the fundamental of K.
- Not that intuitive anymore…

Ocular Dominance Column

- Simplified model



Ocular Dominance Column

- Simplified model: weights matrices for right and left eyes